REMARKS

I. Status of Claims

1. Claims 1, 13, and 25 have been amended.
2. Claims 1-25 are currently pending.

II. Claim Rejections - 35 U.S.C § 103

Claims 1-2, 4-14 and 16-24 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Lucovsky et al., U.S. Patent No. 6,223,207 (hereafter Lucovsky) in view of Sievert et al., U.S. Patent No. 6,687,729 (hereafter Sievert). Claims 3 and 15 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Lucovsky in view of Sievert further in view of Brown et al., U.S. Patent No. 6,631,363 (hereafter Brown). Claim 25 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Lucovsky in view of Sievert further in view of Bhattacharya, "Design Notes on Asynchronous I/O (aio) for Linux" (hereafter Bhattacharya).

Applicants' attorney has amended independent claims 1, 13, and 25 to clarify the steps involved with receiving an alert event at an event port. The claim amendments are fully supported by the application, as filed. For example, the first full paragraph on page 13 of the specification, as originally filed, states that requests are returned immediately to their respective applications, regardless of the events requested or any associated timeouts. Support for the amended limitations can also be found in the first paragraph of page 19 of the specification, as originally filed, which discloses that an alert event is returned alone as a "single" and "only" event by respective application threads, regardless of the events the respective threads were waiting for.

Independent Claim 1 has been specifically amended to further recite the step of returning each computer software application thread to its respective computer software application independent of the events that the computer software application threads were waiting to retrieve. Lucovsky discloses an I/O completion port object for reporting the completion of an asynchronous I/O request to the thread that issued the I/O request in a multi-threaded environment. Sievert discloses a work queue with program controllable states for managing items of work executable by reusable threads. With regards to the prior and amended limitations of Claim 1, it is respectfully submitted that Lucovsky in view of Sievert neither teaches nor suggests at least the limitations of (A) an alert event that is "generated by a computer software application", (B) the changing of a state of the event port "in response to receiving an alert event" and (C) returning at least one computer software application thread to its respective computer software application in a manner that is "independent of an event that the at least one computer application threads is waiting to retrieve".

With regards to (A), it is respectfully submitted that the completion events in Lucovsky are generated by an I/O subsystem (66), not "a computer software application" as claimed in Claim 1 (col. 10, lines 4-7 of Lucovsky). Lucovsky particularly draws a distinction between the applications/clients (50/52) and the subsystems/servers (48) in the computer system (10)(col. 6, lines 40-53). While the two components interact, they are clearly demonstrated in the teachings of Lucovsky to be different entities, especially with regards to performing operations and generating results (col. 7, lines 1-9 and Figure 5). In contrast, the source of the "alert event" in the claimed invention is "a computer software application". This difference between the claimed invention and the elements cited in the reference of Lucovsky is an essential part of a benefit provided by the claimed invention, wherein an alert event generated by a computer software application enables several active threads to synchronously share the same operating system resources in a manner that avoids performance degradation. While the teachings of Sievert mention an IO completion port and thread operations (20, 24), it is respectfully submitted that they do not teach or suggest this particular feature either, especially when this feature is considered in the context of Claim 1 as a whole.

With regards to (B), Sievert teaches that states exist for a work queue, the state of a work queue is initially "stopped", and certain methods are used to change the states of the work queue (col. 3, lines 27-44; col. 6, lines 65-66). However, Sievert is silent with regards to the basis upon which a state is changed using the disclosed methods. In contrast, Claim 1 designates that a state change occurs "in response to a receiving the alert event". The Office Action states that Lucovsky does not teach this limitation either (page 3, lines 4-5). Thus, in view of the lack of connection between an event and an operation state change in the disclosure of Sievert, it is respectfully submitted that Sievert does not teach or at least suggest the above limitation, even when taken in consideration of the primary teachings of Lucovsky.

With regards to (C), it is respectfully submitted that neither Lucovsky nor Sievert suggests returning a computer software application thread "independent of an event that the at least one computer application thread is waiting to retrieve". Lucovsky discloses threads (70) being handed I/O completion packets, as was cited in the most recent Office Action (col. 13, lines 7-11). However, this does not teach or suggest every aspect of the amended limitation cited above.

Both of the disclosed "return" situations in the teachings of Lucovsky rely upon completion information. In the first "return" situation, the application thread (72) returns when there is completed I/O available, based on a GetQueuedCompletionStatus function (col. 10, lines 36-41). This function specifically involves returning bytes read or written for an I/O request and I/O errors that may have occurred with an I/O service request (col. 11, lines 48-67). In the second "return" situation, a call to retrieve completion packets is returned if there are no I/O completion packets available and a timeout value is met (col. 11, lines 63-67). Neither of these operations is "independent of an event that the at least one computer software application thread is waiting to retrieve".

It is respectfully submitted that Sievert does not cure this deficiency either. Sievert discloses the interactions of threads with IO completion ports (col. 7, lines 20-50) and the execution of thread operations (col. 12, lines 8-11). However, this interaction and execution do not teach or suggest the full scope of the above limitation, including the particular feature of performing a return operation in a manner that is "independent of an event that the at least one computer software application thread is waiting to retrieve". As such, it is respectfully submitted that Lucovsky in view of Sievert neither teaches nor suggests all of the limitations as claimed. At least for this reasons, the Applicants' attorney respectfully requests that the pertinent rejections be withdrawn.

The Applicants' attorney also respectfully submits that the amended limitation provides further distinction between an "alert event" and an "an event that the at least one computer software application thread is waiting to retrieve". Specifically, Claim 1 states that the return step is conducted independent of a requested "event", even though an "alert event" is received by the event port. Claim 2 further reiterates this difference by reciting that the return step specifically includes the "alert event". As such, it is respectfully submitted that the "completion information" of Lucovsky cannot be relied upon to teach or suggest the "alert event" of the claimed invention, since the return steps in Lucovsky clearly involve the "completion information" as is discussed in greater detail above.

Independent claims 13 and 25 have been amended in a similar fashion to claim 1. Accordingly, the arguments above with respect to claim 1 can be equally applied to claims 13 and 25. Claims 2-12 and 14-24 depend from amended claims 1 and 13 respectively and are allowable for at least the reason that they depend from an allowable claim. Applicants' attorney respectfully request withdrawal of the above rejections for each of the pending claims.

CONCLUSION

In view of the above amendment, applicants' attorney believes the pending application is in condition for allowance.

Dated: October 25, 2007

Respectfully submitted,

By________________
John W. Branch
Registration No.: 41,633
DARBY & DARBY P.C.
P.O. Box 770
Church Street Station
New York, New York 10008-0770
(206) 262-8900 ▫ (212) 527-7701 fax
Attorneys/Agents For Applicant